REMARKS

Claims 1-4, 11-14, 17-42 were pending in the application. By this paper, Claims 23-42 have been withdrawn without prejudice, Claims 1 and 17 have been amended, and new Claims 43-53 added. Hence, Claims 1-4, 11-14, 17-22, and 43-53 are now pending in the application.

Withdrawn Claims

Pursuant to Par. 1 of the Office Action, Claims 23-42 are herein withdrawn from further consideration without prejudice.

Section 112 Rejections

Per Paragraph 3 of the Office Action, Claim 1, 2 and 20 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, Claim 1 lacked insufficient antecedent basis for the limitation "input strings". Applicant has herein amended Claim 1 to give proper antecedent basis to the previously mentioned limitation, and as such, applicant respectfully submits that the 35 U.S.C. § 112, second paragraph, rejection has been overcome for Claims 1, 2 and 20.

Claim Rejections Under 35 U.S.C. §102

Per Pars. 4-6 of the Office Action, independent Claims 3 and 19 were rejected under 35 U.S.C. §102 as being anticipated by Shapiro (U.S. 4,899,128), and Baird (U.S. 5,848,264), respectively. In response thereto, Applicant provides the following remarks.

Claim 3 – Applicant respectfully disagrees with examiner's contentions that (i) Shapiro teaches "*identifying groups* of said data within said strings that are identical across said plurality of input strings", and (ii) Shapiro teaches "*identifying groups* of said data within said strings that appear in the same order within all of said strings", as asserted by the Examiner in Par. 5 (at the top of page 4) of the Office Action. For both of these propositions (i) and (ii) above, the Examiner cites three lines from the Summary of the Invention section of Shapiro, as follows:

> *"On the other hand, if the comparison succeeds, the two strings may be identical, but it is necessary to perform additional comparisons to establish this. For example, **each** of the elements of the two strings could then be compared."* {Emphasis added}

Applicant sees no inference or suggestion, let alone any explicit teaching, of identifying groups of data for any purpose, let alone for the two distinct and specific purposes enumerated above (and in Claim 3). Shapiro explicitly states that each of the elements of the two strings could be compared (i.e., an element-by-element comparison), but teaches or suggests nothing else; i.e., any sort of identification of groups.

The Examiner's attention is respectfully directed to Col. 2, lines 18-26 of Shapiro, which states:

> *"In accordance with the invention, a comparison of two strings coded with hash codes is then made by first comparing the pointers. If the pointers are the same, then the strings are the same. If the pointers are different, then the hash codes of the two strings are compared. This comparison can readily and quickly be effected in hardware or software. If the comparison fails, it is known that the two strings are not identical and the operating program can move on to the next operation."*

Hence, Shapiro teaches comparing a predesignated portion of the two strings (i.e., "pointers") and then deciding based on this comparison whether to proceed further; i.e., by then comparing a second predesignated portion (hash codes). This can in no way be characterized as "identifying groups", since no "identification" act of any kind is performed. Contrast the foregoing approach of Shapiro with that of Applicant's exemplary embodiment described at page 9 of the specification as filed:

> *"If the excluded set is empty per step 306, groups of contiguous strings which are identical in all inputs are **identified** per step 314. These are referred to as "shared chunks;" shared chunks are described further with respect to Fig. 8 below. In step 316, the **order of the groups** or "shared chunks" is examined to determine if the groups are present in the same order in all inputs. If not, the method 300 returns per step 313. If so, the method 300 continues with step 318 of Fig. 3."*

As previously noted regarding element (ii) above, Applicant's invention of Claim 3 *identifies groups* of data within strings that appear in the same order within all of the strings. Necessary in this process, as reflected in the specification citation above, **is that the groups may be in different orders within the string(s)**. Not only does Shapiro not teach or suggest such identification of groups, but it further teaches away from such analysis of order by specifying predetermined data for comparison (i.e., pointers). No permutation of the order of data within the strings of Shapiro is taught or for that matter even possible; in fact, the invention of Shapiro would not operate as intended if the placement of the data (e.g., pointers) was not precisely where expected.

The Examiner is also referred to the discussion regarding Fig. 8 of Applicant's specification, wherein one specific embodiment of the "identifying groups" methodology of Claim 3 is discussed.

Hence, Applicant submits that Shapiro can in no way render Claim 3 as presented herein anticipated, since not every element thereof is taught or inherent (in fact two separate elements of Claim 3 are completely absent from Shapiro). As such, applicant respectfully submits that Claim 3 is in condition for allowance.

Claim 19 – Applicant respectfully disagrees with examiner's contentions that (i) Baird teaches "*identifying groups* of said data within said first and second strings that are identical across at least both of said strings", and (ii) Baird teaches "*identifying groups* of said data within said strings that appear in the same order within all of said strings", as asserted by the Examiner in Par. 6 (at the top of page 5) of the Office Action. For both of these propositions (i) and (ii) above, the Examiner cites Col 7, lines 28-37 of Baird, as follows:

> *"Debug Trigger Registers*
> *FIG. 6 is a diagram of a debug trigger register for a multi-processor chip. Four debug registers 42 are shared among all three processor cores for triggering debug events. Address field 80 is programmed with a 32-bit address which is compared with addresses generated by the three processor cores. An address match is required for many types of debug events, but not for all types of events.*

For example, an interrupt may trigger a debug event although no address is compared." {Emphasis added}

Applicant sees no inference or suggestion, let alone any explicit teaching, of identifying groups of data for any purpose, let alone for the two distinct and specific purposes enumerated above (and in Claim 19). Baird merely compares two 32-bit addresses for similarity ("address match"), but teaches or suggests nothing else; i.e., any sort of identification of groups or any consideration of their order within data strings. **Applicant specifically requests that the Examiner provide explicit citations to both such instances of "identification of groups" in Baird as claimed in Claim 19.**

Hence, Applicant submits that Baird can in no way render Claim 19 as presented herein anticipated, since not every element thereof is taught or inherent (in fact two separate elements of Claim 19 are completely absent from Baird). As such, applicant respectfully submits that Claim 19 is in condition for allowance.

Claim Rejections Under 35 U.S.C. §103

1. Per Pages 5-10 of the Office Action, independent Claims 1 and 17 were rejected under 35 U.S.C. § 103 as being unpatentable over Kernighan et al. ("The Practice of Programming"), in view of Baird (U.S. 5,848,264). In par. 8 of the Office Action, the Examiner states that Claims 1 and 15-18 are rejected as being obvious under 35 U.S.C. § 103; however, Applicant notes that Claims 15-16 were cancelled without prejudice per Applicant's previous Office Action response. Applicant therefore assumes that this statement was in error and thus only Claims 1, 17-18, and 20 were rejected over Kernighan et al., in view of Baird.

Claim 1 – Applicant respectfully disagrees with examiner's contention that Kernighan et al. teaches "identifying *groups* of said data within said strings", rather Kernighan only teaches "running an old version [of the application program] and the new version for a large number of different test data files, and complains about each *one* for which the outputs are not identical." See Section 6.3, page 149 of Kernighan. Kernighan thus does not teach *identifying groups of said data within strings*, but rather only compares on an individual line-by-line basis (as the Unix

based *cmp* command is only capable of a line by line comparison). There is no identification of groups (such groups that may occur in various different orders within the strings) in Kernighan as is claimed by Applicant, but rather a rote comparison of data files to determine merely if they are identical or not:

> *"It runs the old version (old_ka) and the new version (new_ka) for a large number of different test data files, and complains about each one for which the outputs are* ***not identical****"*

There is simply no (i) identifying groups of data within the strings, or (ii) evaluation of the order of appearance, taught or suggested in Kernighan.

Applicant therefore submits that neither Kernighan nor Baird teach such functionality in combination with the other limitations of Claim 1 and therefore cannot as a matter of law render Claim 1 obvious. Claim 1 is therefore in condition for allowance.

Claim 17 – Applicant respectfully disagrees with examiner's contention that Kernighan teaches "comparing third data strings".

Support for this amendment is provided at, *inter alia*, Page 12, lines 1-13 of Applicant's specification as filed (discussing Fig. 8). As will be noted, this discussion describes a substantially contemporaneous analysis of the data regarding the sequence or ordering of data within all of the inputs:

> *"Next, in step 812, it is determined whether the string before the group (i.e., before the sequence of strings shared among all the inputs) is identical in each of the inputs."*

Stated differently, the process embodiment described above requires that (i) the sequence of strings shared among all of the inputs be identified, and (ii) that the string before the group be evaluated to determine if it is identical in all of the inputs. Hence, all of the inputs must be evaluated in a substantially contemporaneous fashion.

Contrast Kernighan and Baird, wherein only two sets of data (e.g., files or 32-bit addresses) are compared to each other at any given time to generate a decision on a "match", and then a separate comparison must necessarily be performed subsequently. For example, Kernighan at most only teaches the comparison of *two* data strings, as the *cmp* command in Unix is only capable of comparing individual lines of output on *two* files.

Applicant therefore submits that neither Kernighan nor Baird teach such functionality in combination with the other limitations of Claim 17 as amended herein, and therefore cannot render Claim 17 obvious. Applicant respectfully submits that Claim 17 is now in condition for allowance.

2. Per Pages 11-15 of the Office Action, independent Claims 11 and 12 were rejected under 35 U.S.C. § 103 as being unpatentable over Shapiro (U.S. 4,899,128) in further view of Aho ("Compilers Principles, Techniques, and Tools").

Claims 11 and 12 – Similar to the arguments presented in the foregoing discussion of Claim 3, Applicant respectfully disagrees with examiner's contention that Shapiro teaches "*identifying groups* of said data within said strings". Both Claims 11 and 12 include the limitations of (i) "identifying groups of said data within said strings that are identical across said plurality of input strings"; and (ii) "identifying groups of said data within said strings that appear in the same order within all of said strings". The Examiner does not assert that Aho teaches or suggest such limitations (see discussion of Claim 12 on pages 13-14 of the Office Action), and Applicant has previously established that Shapiro does not teach or suggest such limitations. Hence, Claims 11 and 12 cannot as a matter of law be rendered obvious by the combination of these references.

New Claims

By this paper, Applicant has added new Claims 43-54. Support for these new claims is replete throughout the specification as filed (see, *inter alia*, the Example described on pages 13-18 of the specification, as well as the exemplary computer code of Appendix I). Applicant

submits that these new claims each distinguish over the art of record as well, and hence are in condition for allowance.

Other Remarks

Based on the foregoing, Applicant respectfully submits that Claims 1-4, 11-14, 17-22, and 43-54 define patentable subject matter, and are in condition for allowance.

Applicant hereby specifically reserves the right to prosecute claims of different or broader scope in a continuation or divisional application.

Applicant notes that any cancellations or additions made herein are made solely for the purposes of more clearly and particularly describing and claiming the invention, and not for purposes of overcoming art or for patentability. The Examiner should infer no (i) adoption of a position with respect to patentability, (ii) change in the Applicant's position with respect to any claim or subject matter of the invention, or (iii) acquiescence in any way to any position taken by the Examiner, based on such cancellations or additions.

Furthermore, any remarks made with respect to a give claim or claims are limited to only such claim or claims.

If the Examiner has any questions or comments which may be resolved over the telephone, he is requested to call the undersigned at (858) 675-1670.

Respectfully submitted,

GAZDZINSKI & ASSOCIATES

Dated: 5/23/05

By: _____

Robert F. Gazdzinski
Registration No. 39,990
11440 West Bernardo Court, Suite 375
San Diego, CA 92127
(858) 675-1670
(858) 675-1674 (fax)